STUN

The ST User's Network

September 1989

JOH 18 092137B1 09/21/89
NOTIFY SENDER OF NEW ADDRESS
JOHNSON
PO BOX 33188
WRIGHT PATTER OH 45433

STUN

P.O. Box 27285 SLC, Utah 84127-0285

C
PAT JOHNSON
2818 W. 5825 S.
ROY, UT 84067

Officers

President

Neil Breeden

Vice President

Michael Johnson

Secretary

Lee

Board Members

Howard Abrams

Eric Nikolaisen

Brad Davis

Larry Bergan

The ST User's Network is a group dedicated to the advance of Atari 16 bit Computers. We support only the ST series of computers.

Meetings are held the third Wednesday of each month at Murray High, 5300 So. State. General meeting begins at 7:00 pm.

DEADLINES

Issue	Deadline
Oct.'89	Sept. 22, 1989
Nov.'89	Oct. 20,1989
Dec.'89	Nov. 17,1989
Jan. '90	Dec. 22, 1989
Feb. '90	Jan. 19, 1990
Mar. '90	Feb. 16, 1990
Apr. '90	Mar. 16, 1990
May '90	Apr. 20, 1990
Jun. '90	May 18, 1990
Jul '90	Jun. 22, 1990
Aug.'90	Jul. 20, 1990

ADVERTISING RATES

Full Page.....$45.00

Half Page.....$25.00

Quarter Page.....$15.00

Business Card.....$10.00

All quotes single issue rate. Multiple issues receive discount. Club members and non-profit groups call for special rates.

(801)487-8648 Evenings.

STAFF

Editor

Brad Davis

Assistant Editor

Howard Abrams

Advertising Rep

Eric Nikolaisen

STUN is published monthly by the ST User's Network. The opinions and positions expressed are the author's.

Many names used in this publication are trademarks of their respective companies. Their use is for identification only.

STUN welcomes submissions. Please send articles by modem to:
H.O.G. BBS

(801) 964-9764.

Authors without access to a modem, contact the Editor.

(801)487-8648.

STUN

SEPT. 1989

Programming Spreadsheets
Howard Abrams 4

ST Drive Switch
Bill Pike 5

FirmPak ST
Neil Breeden II 6

HiSoft 'C' Interpreter
Howard Abrams 7

Rude Noise

Editorial

Well, a new year begins for S.T.U.N. ....

Neil seems to be going great guns, and after a close shave, we seem to have a pretty solid lineup of officers. I, meanwhile, move from President to Newsletter Editor. Neil certainly is a motivator, since I had every intention of devoting myself entirely to private programming ventures this year. Neil and Howard, however, are my two senior partners in crime. "All for one and one for all" to steal a phrase.

The newsletter returns to monthly printing with this issue, and will remain so for the next year. This means that any support we can get from the membership will be greatly appreciated. Send in the articles!! Reviews, tutorials, hacking hints, anything. Send 'em in...

The newsletter will also be printed on disk beginning with the October '89 issue. Too many reasons to go into here, but if you really want to know them all, give me a call. 487-8648, evenings. Glad to talk to you. Also notice that the entire year's deadlines for ads and submissions appears in the masthead. Will every month, too. Set in concrete, so you can plan the year ahead if you are writing or advertising with us.

One final thing...Check out the ad for DreamPark...Nifty, huh? Look for a totally new game concept from those guys...

Programming Spreadsheets

Howard Abrams

Part II - Keyboard Equivalent Menus

Last time, we talked about using macros to move the cursor around the spreadsheet, to make "bulk-entry" or data easier. In this article, we'll discuss using macros to simulate heavily used menu commands.

INPUT DURING MACRO EXECUTION

Going back to our Checkbook Example from Part I of this series, we could have the program automatically move us from column to column.

In cell AA1, we would begin by typing in {RIGHT}{RIGHT}{RIGHT}. And for ease of use, lets put {DOWN}{LEFT}{LEFT}{LEFT} in cell AA2. This will get us where we want to go, but we need to have the macro stop and allow us to enter figures into the spreadsheet. For this, we put the {?} code before each of the {RIGHT}s. Cell AA1 now looks like: {?}{RIGHT}{?}{RIGHT}{?}{RIGHT}{?}~

We begin by placing the cursor under Check Number, and pressing [ALT]-[Z]. The program will pause and wait for us to type in a value for check number. When we press [RETURN], the program moves our cursor one to the right and waits for us to put in the date, and continues until after amount, which moves us to the beginning of the next line.

We need the tilde ("~") at the end of the line so that when we are finished typing, the program accepts the input and places it in the cell that is highlighted. Otherwise, when the macro finished, we would have to press [RETURN] again. (Actually, this is optional in the current example, because on the next line of the macro, the first command is {DOWN} which make the program accept the input.)

MENUS IN MACROS

When we placed keyboard characters like the arrow keys in a menu, we put a name representation of it in the cell (ie. {LEFT} for the Left Arrow). If we want any other keystrokes, we can just type them in.

For instance, lets suppose that we need to change cells to percentage format with one decimal place. We could write a macro to format any cell that we are pointing to.

First, lets move to cell AA1 and type, "/rfp1~~" (without the quotes, of course). The slash key will get us to the menus, then "r" for Range, "f" for Format, "p" for

continued on page 9...

ST Drive Switch

Bill Pike, Portland Atari Club

If you have a 520ST or 1040ST with a internal drive that drive is always DRIVE A and the external drive is DRIVE B. This is fine until you want to boot a disk from the external drive or the internal drive needs to go the the doctor. You are then up the proverbial estuary without the necessary means of propulsion. Anyhow I have added a switch to my 520ST that will switch the drives making the external drive DRIVE A and the internal drive DRIVE B. This allows you to select which drive will be your boot drive and allows you to operate without a internal drive if necessary. It will also allow you to switch drives, in mid-stream if you have the disk in the wrong drive. You will have to wait until the drive activity light goes out, meaning that the computer is no-longer accessing a drive or you could bomb/kill the disk.

DO NOT ATTEMPT THIS MODIFICATION UNLESS YOU HAVE DONE WORK ON COMPUTER BOARDS BEFORE OR HAVE PREVIOUS EXPERIENCE WORKING ON PRINTED CIRCUIT BOARDS AND IC CHIPS.

HERE IS THE NECESSARY DISCLAIMER: NEITHER THE AUTHOR OR THE PORTLAND ATARI CLUB TAKES ANY RESPONSIBILITY FOR THE RESULTS OF THIS MODIFICATION INCLUDING BOTH SOFTWARE AND/OR HARDWARE DAMAGE IF THIS MODIFICATION IS PERFORMED. THIS MODIFICATION WILL ALSO VOID ANY WARRANTY ON YOUR COMPUTER.

This modification has worked on quite a few computers that it has been used on and the only problem seems to be that some computers might try to write to the disk on boot up but I have had no problem in this area. Anyhow you might want to write protect your boot disks, this is a good idea anyway.

Now to the actual modification. You will need a Phillips Screwdriver, a small slot head screwdriver, a set needle-nose pliers, a set of wire strippers, a set of diagonal cutters,a small soldering iron, de-soldering wick or a solder-sucker, and a 1/4" drill. That concludes the listing of the tools.

Parts required are, a sub-miniature Double Pole/Double Throw switch, and about 3 ft. to number 26 or smaller hookup wire. First the computer selects which drive is going to be used by the YAMAHA (YM2149F)chip labeled U-19. This is the I/O interface chip. The chip sets in the back-center of the board and is next to the RF Modulator which is

continued on page 9...

Firmpak ST

Review

Neil Breeden II

FIRMPAK ST a a small bag containing 95 springs that are placed under the keys on the ST keyboard to stiffen up the keys.

I have just installed the springs and have loaded 1st Word. already I think the keyboard feels better. Although it now sounds slightly different from before it has a better feel. I am told that it is more like the keyboard feel of a Mega, having never typed on a Mega I cannot verify this.

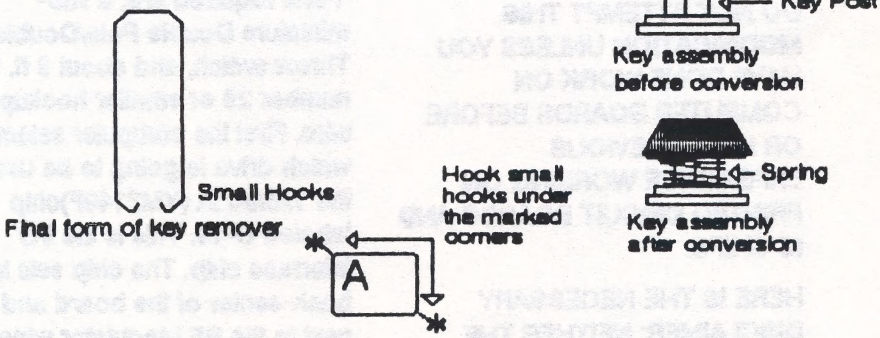
As stated above the kit consists of 95 springs and 1 page of instructions. The docs suggest that the Function keys, the return key and the shift keys not have the springs added under them although enough springs are supplied to add them to these keys if you wish. The instructions also show one method of removing the keycaps from the keyboard. I prefer using a stiff piece of wire. It needs to be about 4 inches long. A large paper clip works well. I bend a small hook on each end of the wire then fold it around my finger. The small hooks are slid up under two opposite corners of the keyboard and the wire pulled up upon. The keycap will then pop off.

Finished layout of the keycap remover:

[See Figure below]

continued on Page 9...

Key Cap Remover



Hisoft 'C' Interpreter

Review

Howard E. Abrams

Do you remember in the "good old days" when you were trying to learn to program in C? "C certainly stands for cryptic" you said as you would type in examples from your book. Then came to moment of truth- compile. I'm sure people three blocks away heard you scream as the errors flowed right off the top of the screen. I remember going back to Basic many times because of my frustration with C.

Those days are over. MichTron is now shipping a C interpreter from HiSoft. Now you can "run" the program immediately like you could with Basic, and compile it later. It has all the abilities that you enjoy with your favorite Basic interpreter, like tracing and variable display, easy to use GEM calls, and a nice GEM-base editor. It even has a memory and stack dump that allows you to view variables in memory.

C IMPLEMENTATION

The version of C that comes with this interpreter uses a K&R version with some UNIX and ANSI function calls. It also includes all the standard AES, VDI, GEMDOS, XBIOS function calls. I noticed that it didn't have the ability to declare immediate constant arrays. For example: int foo[] = { 1, 45, 32, 6, 12, 53, 5, 0, 43, 2, 8}; is not allowed. I also couldn't define aggregates. Structs also posed some problems in that variables defined in one struct couldn't have the same name as another variable in another struct. For example, these two structs couldn't exist in the same program because both x and y are used as variable names in the same structure:

```
struct foo          struct fop
{                   {
int x;               int w;
int y;               int x;
int w;               int y;
int h;               int z;
};                   };
```

To speed up execution of the code, the source code is tokenized in memory. Converting regular ASCII files over is quite easy and is done in the editor itself. It also includes limited help files for reference.

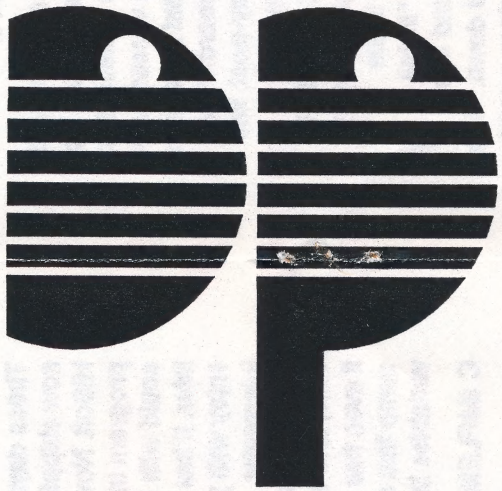
You can break your program up into modules and use header files. Since the header files are tokenized, you must convert your personal ones over. Most

continued on page 12...

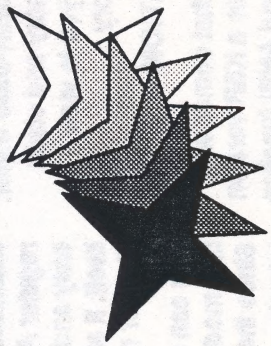
The Transformation has begun

TTTNNN

DreamPark



in New game designs



New in game designs

The Transformation has begun

DreamPark

Programming Spreadsheets

continued from page 5...

Percent, "1" for the decimal place. at this point the program will ask us for the range of cells. Since we just want the particular cell that we have highlighted, we would just press [RETURN], and so we put two "~".

Now name it using the Range-Name-Create "/rnc" command and call it, \P. Now by pressing [ALT]-[P], the program will change the format of the cell that we are pointing to.

INPUT DURING FUNCTIONS

Lets suppose that we would like an easier way of putting in the date in our macro. Sure we could type @DATE(89,6,1) for the date,but after a while, we will begin to look for short-cuts.

Change cell AA1 to read as the following:

{?}{RIGHT}@DATE({?},{?},{?}){RIGHT}{?}{RIGHT}{?}~

Notice that in place of the year, month, and day for the @DATE function, we place {?}. Upon reaching the date column, the program will print on the top line: @DATE(for us. We type in the year and press [RETURN]. The program places a comma, and waits for the month. Now isn't that a little nicer.

AUTO STARTING MACROS

For the sake of completeness, I'm going to introduce the Auto Starting macro. It's purpose probably won't become to useful until after the next article.

If you name a macro "\0", it will automatically go once you've loaded up a sheet with that macro on it. Perhaps if we had a macro that could "find" the next available location for a check, then we wouldn't have to.

Next time we'll explore some codes that one cannot access by any other way than with a macro.

FirmPak ST

continued from page 7...

After all the keycaps are removed it is a good idea to remove all the dust, hair etc that has collected in the keyboard. I use a vaccum with a soft brush attachment to vaccum the dirt from the keyboard. Place the spring over the key-post, line up the correct key over the key-post and press it down till it snaps back into place. The only problem I found was in the instructions. The say "Install a Key-STIFFENER around each key-post, Narrow end down." The narrow end of the springs would not fit over my key-posts. I finally placed the narrow end up and replaced the keys. So far I haven't had a problem with this configuration.

The instructions also state the

following warning:

Although the installation of this product is fairly simple, we strongly recommend that it is done by someone experienced with ATARI hardware.

I had no problem and feel anyone with the desire and an hour to spare should have no problem doing the conversion. NOTE: If you decide to do this upgrade I will not be responsible for your errors.

Well to sum it all up I am very pleased with the FIRMPAK ST upgrade. I program for a living and my keyboard of choice is a VT200 series keyboard. This upgrade makes the keyboard feel a little more like a VT200 series keyboard. I like a stiffer keyboard and the springs really help to stiffen it up.

ST Drive Switch

continued from page 6...

---

a silver-rectangular box located just to the right of the center of the back section of the board. Clear a fairly large area on a table and have good light, you will have things spread out over a fairly large area. #1 disconnect all cables hooked to the computer and turn the computer over on its face with the bottom facing you and the keys next to you. You will see several screws (Phillips head) on the bottom. Remove these screws, three of them will be much longer than the others. Also keep separate, by length, screws that are the same and remember where they came from. #2, slip your hand under the computer and hold the top and the keyboard in place. Then turn over the computer so that it is right-side up. #3, carefully lift the top off the computer and set it aside. You will note that the keyboard is left on the computer. #4, carefully lift the keyboard and note that it is attached to the computer by several wires and a plug on the right hand side. Carefully remove the plug from the computer board and set the keyboard aside. #5, you now to remove the internal drive. There is a metal shield over the cables connecting the drive to the computer on the left side of the drive unit. There are one or two screws holding this small shield, remove these screws and remove the shield and set it aside. You will see a flat ribbon cable, with a plug on it, and several colored wires that go to a plug on the back of the drive. Carefully lift the right hand side of the drive, the side with the disk slot, slightly and then lift the drive and move it slightly to the right. You can then remove the flat ribbon cable by gently pulling on the plug connected to the drive. The other plug has a retainer as part of the jack that it goes into, this is a small pin on the jack that engages a small slot in the plug, lift this with a small slot head screwdriver and

pull the plug gently loose. #6, you now need to remove the Power Supply. There may be a small shield covering the wires connecting the Power Supply to the main board, remove the screws holding this shield and set it and it's screws aside. You may or may not have this shield on your computer. There is a small plug connecting the Power Supply to the main board, remove this plug. You may now lift the power supply clear and set it aside. Older models may not have a internal power supply. #7, you will now see the RF Shield this is held in place by a series of twist clips along the edges. Twist each of these clips to line up with the slot that it is in then gently lift the shield. DON'T FORCE IT. Once you have the shield off set it aside. You are now able to remove the bottom shield by turning the main board over and removing the shield, do so. Turn the board back upright. #8, now locate U-19, this is the fabled YAMAHA chip, this is a 40 pin chip with the label YM2149F on it. Inspect the chip, on one end you will see a dot on one corner, this indicates that the pin, leg, in that corner is number 1. Count down the pins, starting with that pin until you find pins 19 & 20, these should be the last pins on that side of the chip. VERIFY THAT THESE ARE PINS 19 & 20 UNTIL YOU ARE SURE. Now turn the board over and locate the same pins on the bottom of the board. VERIFY THIS UNTIL YOU ARE SURE. #9, BE EXTREMELY CAREFUL AT THIS POINT YOU CAN EASILY LIFT A CONDUCTOR FROM THE BOARD. Now take your soldering iron and the solder-wick or solder-sucker and clean out the holes these pins go thru. Now take the needle nose pliers and gently loosen the pin from each hole. MAKE SURE THAT YOU LEAVE NO SOLDER SPLASHES OR BRIDGES ON THE BOARD. #10, you will now need to drill a hole in the case to put the switch thru. I would suggest a spot between the printer (parallel) port and the external disk drive port. Check that you will have to drill thru the bottom shield just the plastic and that the switch will clear everything. #11, Cut 2 pieces of wire and bare each end. Connect the outside pins of the switch together in the form of a X. Cut 4 lengths of wire 6 inches long and bare both ends. Bare both ends of each piece and solder these pieces to the center terminals and one set of end terminals, it makes no difference which end. Twist the wires that go to each pair of terminals together for identification. #12, install the switch. #13, now back the the chip. Using the needle-nose pliers, from the top of the board gently lift the two pins (19 & 20) clear of the board and bring them out from the side of the chip. BE VERY CAREFUL YOU DON'T BREAK A PIN. DON'T BEND THEM TO MUCH, JUST ENOUGH TO CLEAR THE BOARD AND

HAVE ROOM TO REACH THE HOLES THEY CAME OUT OF. #14, cut the wires from the switch to reach to the pins of the U-19 chip and strip the ends. Then take one pair of wires and insert them thru the open holes, that you removed the pins from. Take the other pair of wires and solder them directly on the pins of the chip. Turn the board over and solder the wires that you poked thru the holes. Neaten up the wires. VERIFY ALL CONNECTIONS AND THAT NOTHING IS TOUCHING ANYTHING IT SHOULDN'T BE ALSO THAT THERE ARE NO SOLDER BRIDGES OR SPLASHES. THIS COMPLETES THE MODIFICATION. #15, Repeat steps #7 thru #1 IN REVERSE ORDER to reassemble the computer. #16, Connect the external drive to the computer and also connect the power and monitor. If the computer fails either of the next tests something is wrong, shut the computer down and re-check what you did. Turn the computer on and verify that the activity light on one of the drives comes on and that the screen display is normal. Turn the computer off and change the drive select switch. Turn the computer back on and verify the other drive activity light comes on and that the screen display is normal. If these tests are OK you have correctly installed the switch.

Hisoft 'C' Compiler

continued from page 8...

standard header files are included. I was surprised to find that it has the ability to "link" in assembly object files in real-time. It is somewhat limited however.

EDITOR

The editor is quite nice with standard features of find and replace (with wildcards), keyboard commands, macros, and the like. It seems to scroll quite quickly. I did notice that I could define a block with the mouse, although I could position the cursor with it.

To pagedown, one can't hit Ctrl-Down Arrow, which I didn't like. All my word processors have that keystroke. Instead, it is F10.

COMPILING

Taking code from the HiSoft interpreter to your compiler . Ierelatively simple. First, covert it from its tokenized format to an ASCII text file, convert the header files, and compile it directly.

The only problem that I encountered was using functions that were HiSoft C specific functions. You see, in order to make GEM commands easier to work with, they have written functions to manipulate windows, make resource dialog boxes, etc. The documentation said that the source code to these functions were included in some text files on the disk, and that one could just

compile them directly into your compiler. I found that only some of these routines were included. I called MichTron to find out if the source code to these functions were available elsewhere, but support from MichTron was extremely poor; not only did the tech not know anything about the program, he didn't even have a manual.

Taking my source code that I've written using Mark Williams and trying it out in interpreted mode was next to impossible, since most of my code uses the commands that I listed above as not being implemented. Although, simple code I'm sure would work.

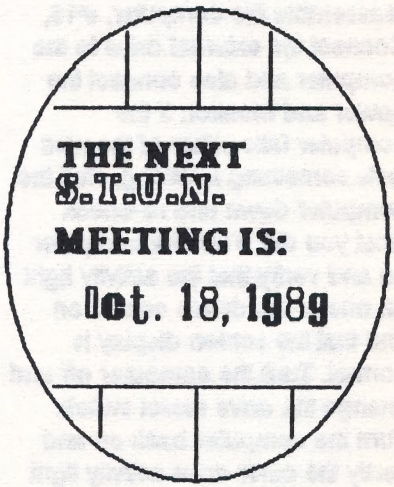
EVALUATION

Although I have listed many of faults with this program, I certainly recommend it. If you are just learning C, there is nothing better than this program and a good tutorial. If you have been developing for many years, I would recommend it because of the decrease in development time spent in compiling.

It would be nice if they would make an ANSI compatible version, but the current version is quite adequate for most needs.

REMEMBER!

Beginning with the next issue (Oct. '89), STUN will be converting to a disk based format, and will be distributed at the monthly meetings. If you aren't able to get your newsletter at the meeting, you may still obtain it two ways:

1. Newsletter issues will be upload to the H.O.G. after each monthly meeting.

2. Copies will be available from the club library for the normal $3.00 charge.